A close up of text

Description automatically generated

ΑΘΗΝΑ 8 Νοεμβρίου 2024

**7η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**

**ΓΙΑ ΤΟ ΜΑΘΗΜΑ “Εργαστήριο Μικροϋπολογιστών”**

**ΟΜΑΔΑ 23**

**Συνεργάτες**

Νικόλαος Αναγνώστου Νικόλαος Λάππας

03121818 03121098

**Ζήτημα 7.1:** Στην συγκεκριμένη άσκηση κληθήκαμε να υλοποιήσουμε σε γλώσσα C τις συναρτήσεις που δίνονται σε assembly για την επικοινωνία του μικροελεγκτή ATmega328PB με τον αισθητήρα θερμοκρασίας DS1820/DS18B20. Θα πρέπει να προσέχουμε ότι κάνουμε χρήση τον ακροδέκτη PD4 και να τροποποιούμε μόνο αυτόν όπως ακριβώς ορίζουν οι συναρτήσεις, και πως σε αυτόν τον ακροδέκτη να είναι συνδεδεμένος μόνο ένας αισθητήρας. Ζητείται επιπλέον μια συνάρτηση, την ονομάσαμε και είναι τύπου int16\_t, η οποία χρησιμοποιεί τις συναρτήσεις που δίνονται σε assembly και επιστρέφει την τιμή που κατέγραψε ο αισθητήρας. Ακολουθούν οι συναρτήσεις με αναλυτικούς κώδικες:

#define F\_CPU 16000000UL

#include<avr/io.h>

#include<avr/interrupt.h>

#include<util/delay.h>

#include<stdbool.h>

// Returns true if a connected device is found (PD4 = 0)

bool one\_wire\_reset()

{

DDRD |= (1 << PD4); // Set PD4 as output

PORTD &= ~(1 << PD4); // Clear PD4

\_delay\_us(480); // Delay 480 usec

DDRD &= ~(1 << PD4); // Set PD4 as input

PORTD &= ~(1 << PD4); // Disable pull-up resistor

\_delay\_us(100); // Delay 100 usec

uint8\_t input = PIND & (1 << PD4); // Read input

\_delay\_us(380); // Delay 380 usec

// If device is detected (PD4 = 0) -> return true

if (input == 0x10) {return false;} // PD4 = 1

return true; // PD4 = 0

}

uint8\_t one\_wire\_receive\_bit()

{

DDRD |= (1 << PD4); // Set PD4 as output

PORTD &= ~(1 << PD4); // Clear PD4

\_delay\_us(2); // Delay 2 usec

DDRD &= ~(1 << PD4); // Set PD4 as input

PORTD &= ~(1 << PD4); // Disable pull-up resistor

\_delay\_us(10); // Delay 10 usec

uint8\_t bit\_to\_receive = (PIND & (1 << PD4)) ? 1 : 0;

\_delay\_us(49); // Delay 49 usec

return bit\_to\_receive;

}

void one\_wire\_transmit\_bit(uint8\_t bit\_to\_transmit)

{

DDRD |= (1 << PD4); // Set PD4 as output

PORTD &= ~(1 << PD4); // Clear PD4

\_delay\_us(2); // Delay 2 usec

//PORTD |= (bit\_to\_transmit & 0x10); // Send PD4 bit to connected device

PORTD = (PORTD & ~(1 << PD4)) | ((bit\_to\_transmit & 0x01) ? (1 << PD4) : 0);

\_delay\_us(58); // Delay 58 usec

DDRD &= ~(1 << PD4); // Set PD4 as input

PORTD &= ~(1 << PD4); // Disable pull-up resistor

\_delay\_us(1); // Delay 1 usec

}

uint8\_t one\_wire\_receive\_byte()

{

uint8\_t received\_byte = 0x00; // Store the byte (8-bit) we received

for (uint8\_t i = 0; i < 8; i++)

{

uint8\_t received\_bit = one\_wire\_receive\_bit();

received\_byte |= (received\_bit << i); // Logical shift left, because DS18B20 send LSB first

// Logical OR to insert new bit into byte sequence

}

return received\_byte;

}

void one\_wire\_transmit\_byte(uint8\_t byte\_to\_transmit)

{

for (uint8\_t i = 0; i < 8; i++)

{

uint8\_t send\_bit = (byte\_to\_transmit >> i) & 0x01;// Bit to transmit now in position bit 0

one\_wire\_transmit\_bit(send\_bit);

}

}

int16\_t GetTemperature()

{

bool connected\_device = one\_wire\_reset(); // Check for connected device

if (!connected\_device) return 0x8000; // Error in connection return 0x8000

one\_wire\_transmit\_byte(0xCC); // Only one device

one\_wire\_transmit\_byte(0x44); // Begin counting temperature

while (!one\_wire\_receive\_bit()); // Wait until the above counting terminates

one\_wire\_reset(); // Re-initialize

one\_wire\_transmit\_byte(0xCC);

one\_wire\_transmit\_byte(0xBE); // Read 16-bit result of temperature value

uint16\_t temperature = 0;

temperature |= one\_wire\_receive\_byte(); // 8-bit LSB of the total 16-bit value

// Shift the 8-bit value 8 times to the left, OR with the previous 8-bit value

// And take the temperature value of 16-bit

temperature |= ((uint16\_t)one\_wire\_receive\_byte() << 8);

return temperature;

}

**Ζήτημα 7.2:** Σε αυτή την άσκηση κληθήκαμε να χρησιμοποιήσουμε τις συναρτήσεις που κατασκευάσαμε προηγουμένως και να εντάξουμε την εκτύπωση της θερμοκρασίας. Συγκεκριμένα, χρησιμοποιήσαμε τον τρόπο με το PCA9555 προκειμένου να αποφύγουμε οποιαδήποτε ατασθαλία, αφού τόσο το LCD όσο και το θερμόμετρο επηρεάζει το PORTD και το PD4 αντίστοιχα. (Θα μπορούσε να γίνει το LCD κανονικά και με το PORTB, απλά με προσοχή όταν γίνεται χρήση του PD4 από τον αισθητήρα θερμοκρασία να μην κάνουμε εκτύπωση με το LCD). Παρακάτω ακολουθεί η συνάρτηση διότι οι υπόλοιπες συναρτήσεις είναι της 7.1 και το LCD με PCA υπάρχει ήδη σε προηγούμενη σειρά:

int main(){

DDRB = 0xff;

DDRC = 0x00;

twi\_init();

PCA9555\_0\_write(REG\_CONFIGURATION\_0, 0x00); // EXT\_PORT0 -> output

lcd\_init();

while(1)

{

lcd\_clear\_display();

uint16\_t temperature = GetTemperature();

if (temperature == 0x8000) // NO Device 9 bits no need for extra line

{

lcd\_data('N');

lcd\_data('O');

lcd\_data(' ');

lcd\_data('D');

lcd\_data('e');

lcd\_data('v');

lcd\_data('i');

lcd\_data('c');

lcd\_data('e');

}

else {

if ((temperature & 0x8000) > 0) {

temperature = ~temperature + 1;

lcd\_data('-');

} //if negative convert to its value

else lcd\_data('+');

int dekadika = 0;

//for now

if((temperature&0x01)==0x01) dekadika += 625;

temperature = temperature>>1;

if((temperature&0x01)==0x01) dekadika += 1250;

temperature = temperature>>1;

if((temperature&0x01)==0x01) dekadika += 2500;

temperature = temperature>>1;

if((temperature&0x01)==0x01) dekadika += 5000;

temperature = temperature >> 1 ;

int result = 0;

for (int i = 0; i <= 6; i++){

if (temperature & 0x0001 > 0) result += (int)pow(2,i);

temperature = temperature >> 1;

}

bool is\_zero = false;

if(result/100!=0) {

lcd\_data('0'+(result/100));

result = result % 100;

}

else is\_zero = true;

if(result/10==0 && is\_zero==true) ;

else{

lcd\_data('0'+(result/10));

result = result % 10;

}

lcd\_data('0'+result);

//might add something here

lcd\_data('.');

result = dekadika/1000;

lcd\_data('0'+result);

dekadika %= 1000;

result = dekadika/100;

lcd\_data('0'+result);

dekadika %= 100;

result = dekadika/10;

lcd\_data('0'+result);

dekadika %= 10;

lcd\_data('0'+dekadika);

lcd\_data(' ');

lcd\_data(223);

lcd\_data('C');

}

\_delay\_ms(750);

}

return 0;

}